Remarks

Claim rejections 35 USC § 101

Rejection of claims 11 and 12 as being directed towards non-statutory subject matter.

As suggested in section 1 of the Office Action, claims 11 and 12 have been amended to positively recite that the computer readable medium of the computer program product is non-transitory. In view of this amendment, Applicants request that Examiner withdraw the rejection of claims 11 and 12 under 35 USC § 101.

Rejection of claim 14 as being directed towards non-statutory subject matter.

Claim 14 has been amended to positively recite a computer, and that the computer comprises the request generator, the network connection, the comparison means, and the routing instruction.

In view of this amendment, Applicants request that Examiner withdraw the rejection of claim 14 under 35 USC § 101.

Rejection of claims 15 and 16 as being directed towards non-statutory subject matter.

Claims 15 and 16 have been redirected towards computer program product claims, wherein the computer program product comprises a non-transitory computer-readable medium storing and/or recording instructions in machine readable form which when executed in a computer system for managing contacts at a contact centre are effective to cause the computer system to create a software object having the features of previous claim 15, and a virtual queue of contacts having the features of previous claim 15, respectively. Applicants request that Examiner withdraw the rejection of claims 15 and 16 under 35 USC § 101.

Claim rejections 35 USC § 103

Rejection of claims 1 to 3, 5 to 7, 11, 13 and 17 as being unpatentable in view of Flockhart et al, Webber and Goodrich.

Claim 1 has been amended by deleting the feature of "whereby the received contact can be prioritised relative to other ones of said contacts" and more positively reciting the determination of the queuing position for the new software object as being determined from a comparison of the priority of the received contact and a priority of the contact represented by the at least one other software object.

Claim 1 is directed towards a method of managing contacts in a contact center. The method involves assigning to contacts received at the center, a priority and a skillset identifier. A software object is created for the contact and placed in a prioritized queue for a skillset, with at least one other software object representing a contact with a similar skillset identifier to that of the received contact.

The queuing position of the software object is determined by comparing the priority of the contact with that of the contact represented by the at least one other software object. Thus, at any one time, a software object at a 'head' of the prioritized queue is determined as representing the next contact to be selected for processing by an agent of the contact center, whereas a software object at a 'tail' of the prioritized queue is determined as representing the last contact, of the received contacts, to be selected for processing by an agent of the contact center.

The software object is provided with a pointer to the at least one other software object. Thus, each software object within the prioritized queue can have information about contacts immediately ahead and/or behind it in the prioritized queue so that each software object "knows" its place in the prioritized queue, thereby eliminating any requirement for a contact manager to manage a list of the software objects.

Flockhart, (Flockhart et al, US 6,535,601) is concerned with improving call handling in a contact center, and in particular, ensuring that received contacts, which were

assigned a low priority, are not being kept waiting indefinitely in a prioritized queue, as long as there is a contact having a higher priority to be handled, [column 1, lines 57 to 67].

To this end, Flockhart discloses a call center system arranged to receive and handle call contacts. On receipt of contacts at the call center, the system is arranged to determine a skill requirement associated with each of the contacts, [abstract, and Fig.1A].

The contacts are then placed in a suitable skill queue, in the order of their arrival at the system, [Fig. 2, step 200, and column 4, lines 62 to 65].

The system assigns to each of the contacts, a priority level for access to the associated skill. In particular, each contact is tagged with a value indicative of the priority level, i.e., HIGH, MID, or LOW, [Fig. 2, step 202, column 5, lines 3 to 5].

The system then identifies the first occurrence of HIGH, MID and LOW value contacts in the queue, [Fig. 2, step 206].

An advantage adjustment is then applied to the contacts based on the priority value, [Fig. 2, step 204, 208, column 5, lines 19 to 20]. In one embodiment, the adjustment time is calculated as a sum of an actual wait time, i.e., the time that has already elapsed, and an advantage adjustment, [column 5, line 66 to column 6, line 1]. In a second embodiment, the adjustment time is calculated as a sum of a predicted weight time, which includes an actual wait time and a weighted advance time, and an advantage adjustment, [column 6, lines 1 to 5].

The adjustment time of the advantage adjustment for each contact is dependent on the priority value associated therewith. As exemplified in Fig. 4, a contact tagged with a priority level of HIGH is assigned an adjustment time of +20 seconds, a contact tagged with a priority level of MID is assigned an adjustment time of +10 seconds, and a contact tagged with a priority level of LOW is assigned an adjustment time of 0 seconds, [Fig. 4, column 5, lines 32 to 38].

The contact in the queue having the greatest adjustment time of the contacts in the queue is then moved to the 'call selection pool' from where a suitable agent selects the contact for processing, [Fig. 4, step 210].

The system traverses the queue from a head position to determine the greatest adjustment time, [column 6, lines 45 to 50]. It will be appreciated that if the contact at the head of the queue has a priority value of HIGH, that contact will of course then have the greatest adjustment time. However, if the contact at the head of the queue has a priority value of LOW and is followed in the queue by a contact having a priority value of HIGH, the LOW priority contact may nonetheless be selected for processing based on it being associated with an actual wait time that exceeds 20 seconds longer than the actual wait time of the contact having the HIGH priority level.

In summary, received contacts are placed in skill queues based on their skill requirements, in the order they are received at the contact center and are each assigned a priority value. An advantage adjustment is applied to each contact in the queue and the next contact having the greatest adjustment time is the next contact selected for processing by an agent.

As with Flockhart, the method of the present invention involves assigning to the contact, a priority and a skillset identifier, and selecting and queuing the contact in a queue based on the required skillset.
However, and in contrast to Flockhart, the software objects representing the received contacts of the present invention are inserted into a queue at a queuing position determined from a comparison of the priority of the received contact and a priority of the contact represented by the at least one other software object, i.e., a prioritized queue.

The contacts are not placed in the queue in an order they are received at the contact center as is the case with Flockhart.

As acknowledged in the Office Action at paragraph 4, page 4, Flockhart does not disclose adding to new software objects, pointers to the at least one other software object in the queue.

As can be understood from section 5, B, page 9 of the Office Action, Flockhart does not disclose assigning to the received contact, multiple skillset identifiers, and adding to software objects representing the contact in different queues, separate pointers.

Thus, in summary, Flockhart does not disclose the following features of claim 1:

- determining the queuing position from a comparison of the priority of the received contact and a priority of the contact represented by the at least one other software object;
- adding to the new software object a pointer to the at least one other software object; and
- the step of assigning to the received contact the priority and the skillset identifier including assigning multiple skillset identifiers, and wherein the step of adding the pointer comprises adding separate pointers to software objects in different queues.

Webber (Webber, US 6,883,006) is concerned with circular linked lists and with reference to Fig. 3A to 3C, discloses a system (300) arranged to add an element or member (302) to an existing circular singly linked list (301). Each element or member (302) of the linked list (301) comprises a content field (304) and a next pointer field (306). The next pointer filed contains a pointer to a next member (302) in the circular singly linked list (301). The system further includes a register (308) for storing a next pointer associated with a current member of the list. A new member can be inserted into the circular singly linked list (301) by updating the pointer of a member positioned immediately before the desired position of the new member to point to the new member, and by including in the pointer field of the new member, the pointer previously contained in the next pointer field of the member positioned immediately before the desired position of the new member.

Thus, Webber teaches that entries in a list or a queue may include fields containing pointer to a next element in the queue, thereby providing linked lists.

The Office Action alleges in paragraph 6 of page 4, that the skilled person would modify the queue of Flockhart to employ a linked list of software objects including

pointers, as disclosed in Webber, in order to add entries dynamically as the queue of Flockhart is required to grow. Reconsideration is requested.

The contacts of Flockhart are placed in the queue in the order they are received at the contact center. Thus, were the teachings of Flockhart and Webber combined in the manner suggested, each new contact being added to the queue would be adapted to include a pointer to the next contact, i.e. the penultimately received contact, which has not yet been selected for movement to the selection pool.

Even if this were the case, it would not provide the advantages of the present invention in that the system would still need to maintain a list of the entries in order to be able to ascertain which contact is the next contact for movement to the selection pool. The contact would not "know" how it ranks relative to the other contacts in the list of contacts waiting to be handled by an agent of the contact center because the contacts are not queued in the order in which they will be dealt with. The system will still need to maintain a list of the contacts including an arrival time for each contact in order to be able to select the contact having the greatest adjustment time and thus the contact for movement into the selection pool.

Furthermore, Webber does not disclose the features of:

- determining the queuing position from a comparison of the priority of the received contact and a priority of the contact represented by the at least one other software object; and
- the step of assigning to the received contact the priority and the skillset identifier including assigning multiple skillset identifiers, and wherein the step of adding the pointer comprises adding separate pointers to software objects in different queues.

Goodrich, (Goodrich et al, "Lists and Sequences"), discloses a number of methods of storing and manipulating List ADTs (abstract data types), and in particular, that an object can point any other arbitrary object as part of the same simple list.

The Office Action relies upon Goodrich to provide evidence of multi-dimensional linked lists or trees and alleges that the skilled person would be motivated by this

reference to further modify the teaching of Flockhart in combination with Webber to assign multiple skillset identifiers to a received contact and furthermore to add to a software object representing the contact, separate pointers to software objects in different queues. Reconsideration is also requested.

Firstly, Goodrich does not suggests or discloses "multidimensional linked lists". All the teaching relates to simple lists, either singly or doubly linked. Furthermore, nothing in the document suggests implementing links between an element in a first list and an element in a second list.

Accordingly, Applicants refute the suggestion that this document is evidence of common knowledge of a pointer pointing to any other object "to create a data structure such as a multidimensional linked list or tree", when clearly it makes no mention of such structures.

Furthermore, nothing in Goodrich teaches or suggests assigning multiple skillset identifiers to a contact, which is also not suggested in Flockhart or Webber. This claim feature cannot have been obvious even if (as is disputed for the reasons given above) the abstract concept of what the Office Action refers to as "multidimensional linked lists" was part of the common knowledge.

Furthermore, nothing in the prior art teaches the desirability of linking between skillset queues, even if the abstract concept of what the Office Action refers to as "multidimensional linked lists" was part of the common knowledge (again this is disputed for the reasons given above).

Summary

Despite the allegations of the Office Action, the fact remains that the method of claim 1, as amended, now requires the feature of determining the queuing position from a comparison of the priority of the received contact and a priority of the contact represented by the at least one other software object.

The queue of Flockhart is not a prioritized queue in that contacts are placed in the queue in the order that they arrive at the contact center. In fact, Flockhart teaches away from the prioritization of a queue, [column 3, line 63, to column 4, line 1].

Thus, even if the teachings of Flockhart, Webber, and Goodrich or common general knowledge were combined, the skilled person would not arrive at the invention as claimed as none of the cited references disclose all of the features of claim 1.

Claim 1 is thus submitted to be novel and non-obvious in view of a combination of Flockhart, Webber, and Goodrich.

Independent claims 11 and 13 have been brought into conformity with amended claim 1, and are therefore also submitted to be novel and non-obvious in view of a combination of Flockhart, Webber, and Goodrich.

Claims 2, 3, 5 to 9, and 17 are dependent on claim 1, and are therefore also submitted to be novel and non-obvious in view of a combination of Flockhart, Webber, and Goodrich.

Rejection of claims 8, 10, 12 and 14 as being unpatentable in view of Flockhart et al. and Webber.

Claim 8 has been amended to recite that the contact has an assigned priority, and furthermore, that the software objects containing pointers provides a prioritized queue, wherein each software object is positioned in the prioritized queue in order of priority with respect to the one or more other software objects.

As argued above with respect to claim 1, a combination of Flockhart and Webber would not lead the skilled person to order the software objects within the queue based on the priority assigned to the contact which they represent since the contacts in the queue of Flockhart are placed in the queue in the order that they are received. Furthermore, Flockhart teaches away from the ordering of contacts within a queue on the basis of priority due to a disadvantage that may affect low ranked contacts

waiting indefinitely within a queue due to the constant influx of contacts assigned a higher priority. Webber does not mention the prioritizing of contacts within a queue.

Thus, even if the teachings of Flockhart and Webber were combined, the skilled person would not arrive at the invention as claimed as none of the cited references disclose all of the features of claim 8.

Claim 8 is thus submitted to be novel and non-obvious in view of a combination of Flockhart and Webber.

Independent claims 10, 12 and 14 have been brought into conformity with amended claim 8, and are therefore also submitted to be novel and non-obvious in view of a combination of Flockhart and Webber.

Claim 9 is dependent on claim 8 is therefore also submitted to be novel and non-obvious in view of a combination of Flockhart and Webber.

Rejection of claims 15 and 16 as being unpatentable in view of Flockhart et al, and Webber.

Claims 15 and 16 have been amended to recite that the contacts have an assigned priority and the software objects are positioned in prioritized skillset queues in order of priority with respect to the one or more other software objects.

As previously argued in respect of independent claims 1 and 8, a combination of Flockhart and Webber would not lead the skilled person to order the software objects within the queue based on the priority assigned to the contact which they represent since the contacts in the queue of Flockhart are placed in the queue in the order that they are received. Furthermore, Flockhart teaches away from the ordering of contacts within a queue on the basis of priority due to a disadvantage that may affect low ranked contacts waiting indefinitely within a queue due to the constant influx of contacts assigned a higher priority. Webber does not mention the prioritizing of contacts within a queue.

Thus, even if the teachings of Flockhart and Webber were combined, the skilled person would not arrive at the invention as claimed as none of the cited references disclose all of the features of claims 15 or 16.

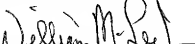
Claims 15 and 16 are therefore submitted to be novel and non-obvious in view of a combination of Flockhart and Webber.

In view of the amendments and arguments made herein, the applicants respectfully request the examiner withdraw the rejections, and allow the application.

A needed Petition For Extension of Time is also submitted herewith.

January 18, 2011

Respectfully submitted,



William M. Lee, Jr.
Registration No. 26935
Barnes & Thornburg LLP
P.O. Box 2786
Chicago, Illinois 60690-2786
(312) 214-4800
(312) 759-5646 – Fax